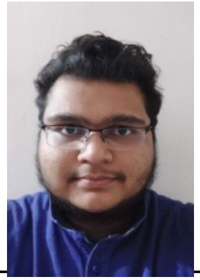
Machine Learning and Applications (UE20EC352)

Final Project Submission

Domain: Networking

Delay Estimation of various parameters in mobile traffic environment using Time series Datasets.

Team

Gautham Bolar	**PES1UG20EC044**	
Chennamsetti Sai Pranay	**PES1UG20EC048**	
Dheemanth R Joshi	**PES1UG20EC059**	

Motivation

- Delay / Latency estimations is one of a major problems faced in the field of Edge Computing environments.
- Channel bandwidth allocation, task scheduling and service migration are some of the critical threads performed by an edge computing system which depends on the delay constraints.
- However, due to highly uncertain and dynamic environments accurate delay estimations becomes challenging.

Traffic Prediction Dataset

Datasets

Metro Interstate Traffic Volume Data set

Image Recognition Task Execution Times In Mobile Edge Computing

Dataset 1: Traffic Prediction Dataset

Description	**This dataset contains the number of cars passing through four junction measured at an hourly frequency. The measurements are taken over the course of nearly two years (from 2015-11-01 to 2017-06-30).**
Input Features	**Date Time, Junction Number**
Output Features	**Traffic Density in the future time slots**
Significance	**By predicting Traffic densities, the efficiency of traffic management can be significantly increased.**

Dataset Link: https://www.kaggle.com/datasets/fedesoriano/traffic-prediction-dataset?resource=download

Dataset 2: Image Recognition Task Execution Times In Mobile Edge Computing

Description	**This dataset contains execution times for offloaded image recognition tasks that were performed in a mobile edge computing environment. The tasks were offloaded from a client device (mobile edge node) to one of several edge servers, and the execution times were recorded for each server.**
Input Features	**Time: day, date, hours, minutes, second, year**
Output Features	**Turnaround Task Execution time: in seconds**
Significance	**By learning the trend of the execution time, critical processes like task partitioning and bandwidth allocation can be handled efficiently**

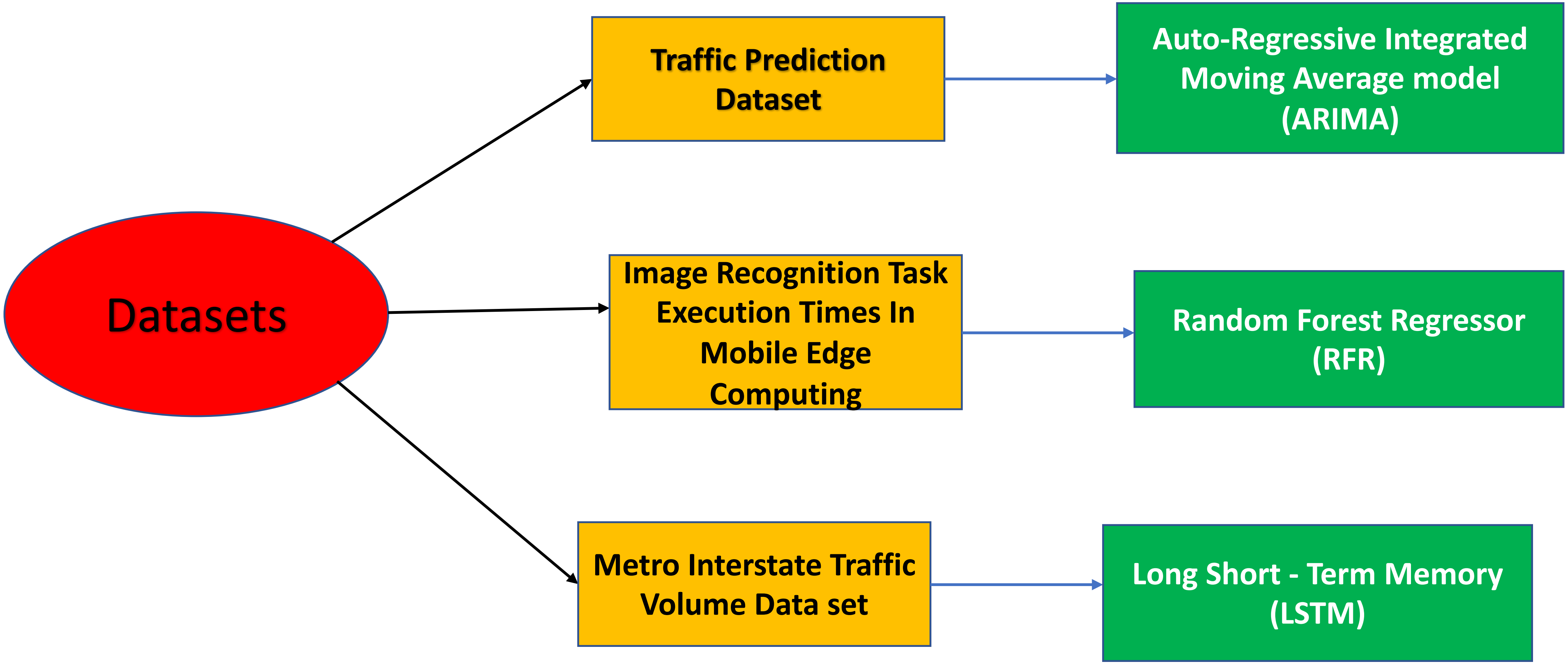
Dataset Link:
https://archive.ics.uci.edu/ml/datasets/Image+Recognition+Task+Execution+Times+in+Mobile+Edge+Computing

Dataset 3: Metro Interstate Traffic Volume Dataset

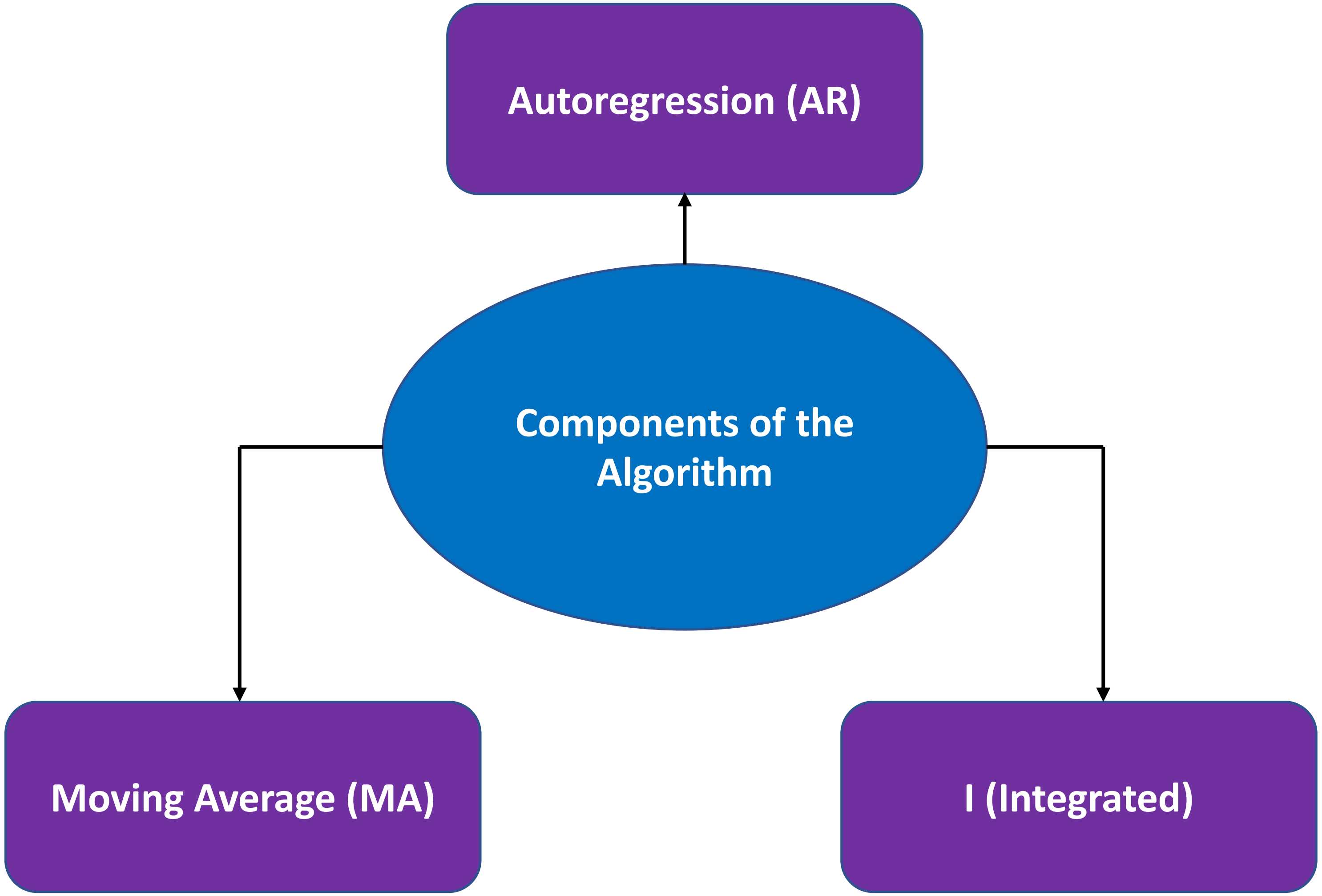
Description	**This Metro Interstate Traffic Volume dataset contains hourly traffic volume data for the I-94 Interstate highway in the Minneapolis-St. Paul metropolitan area of Minnesota, USA. The dataset was collected by the Minnesota Department of Transportation from 2012 to 2018, and includes 48,204 observations.**
Input Features	**13 Input features Including:**
Output Features	**Hourly Traffic Volume on The I-94 Highway**
Significance	**By predicting Traffic densities, the efficiency of traffic management can be significantly increased.**

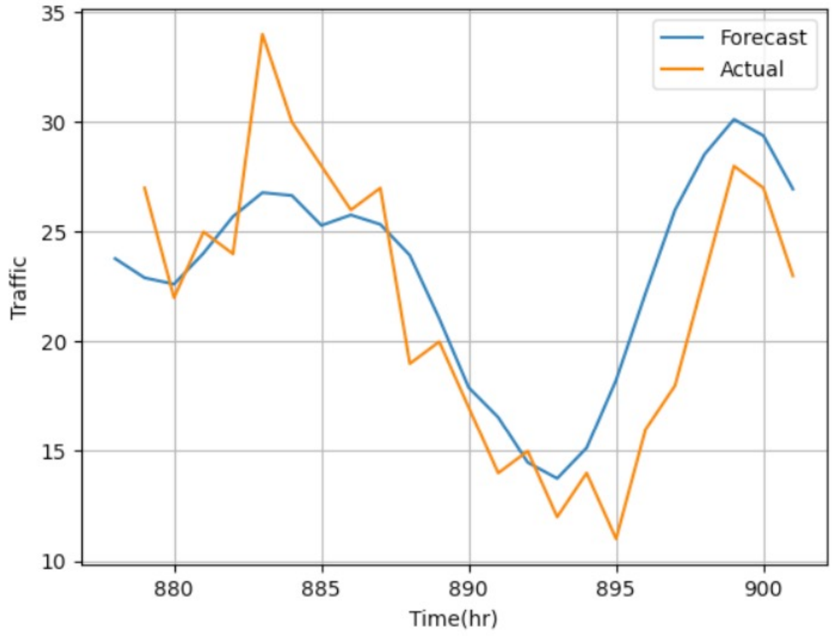
Dataset Link: https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume

Machine Learning Algorithms used for the datasets



Auto-Regressive Integrated Moving Average model (ARIMA)





The ARIMA model is specified using three parameters: p, d, and q. The parameter p is the order of the AR component, d is the degree of differencing required to make the time series stationary, and q is the order of the MA component.

Random Forest Regressor (RFR)

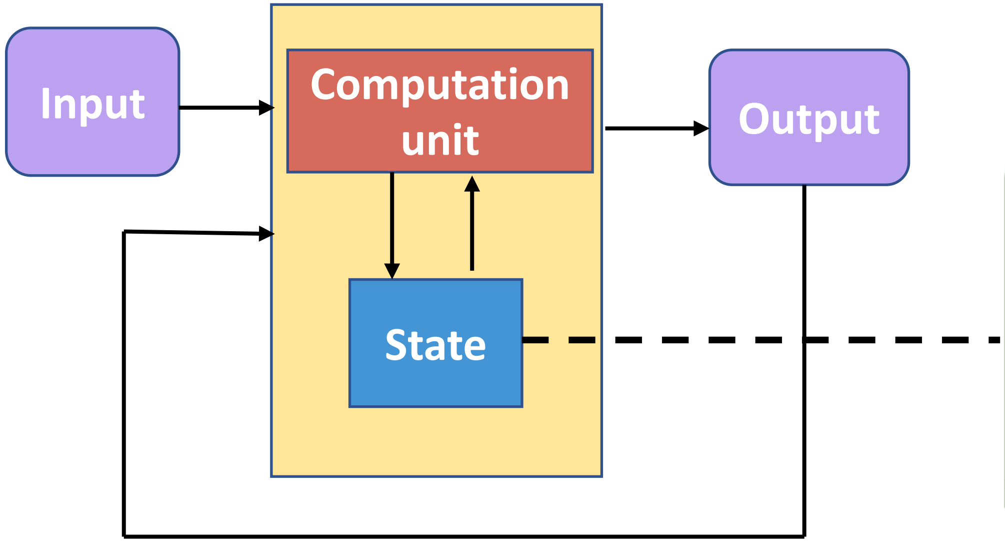
- **Random Forest Regressor is a supervised machine learning algorithm used for regression tasks. It is based on the concept of an ensemble of decision trees, where each tree is trained on a random subset of the data and a random subset of the features. During the training process, each tree makes a prediction for the target variable based on the input features, and the predictions from all trees are combined to generate the final prediction.**

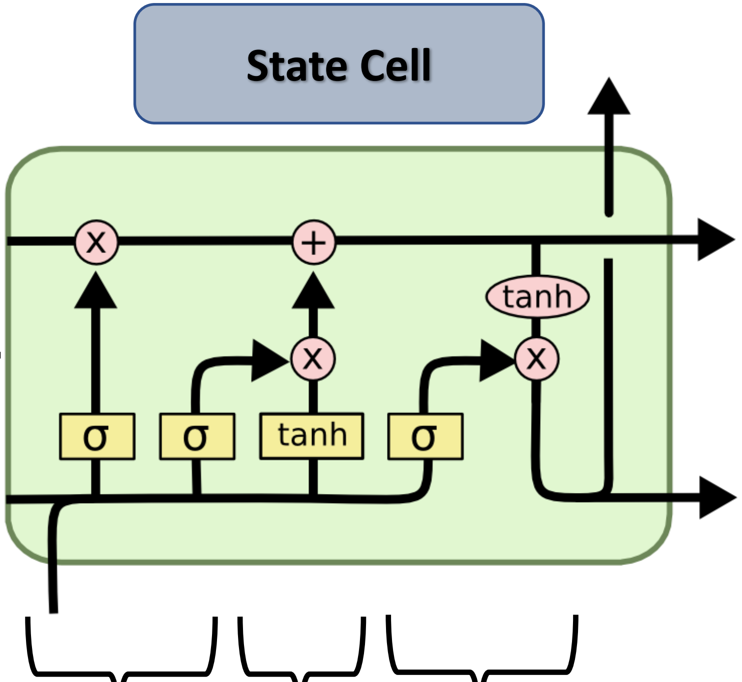
Can Handle both linear and non linear relationships

Features of RFR

Robust against Overfitting

Algorithm 3: Long Short - Term Memory (LSTM)





1) LSTMs have feedback loops
2) Experience is maintained by a cell
3) Ideal for highly random and large datasets

Forget Gate

Input Gate

Output Gate

All the gates have range between [0,1] which indicates the portion of forgetting, input and output respectively.

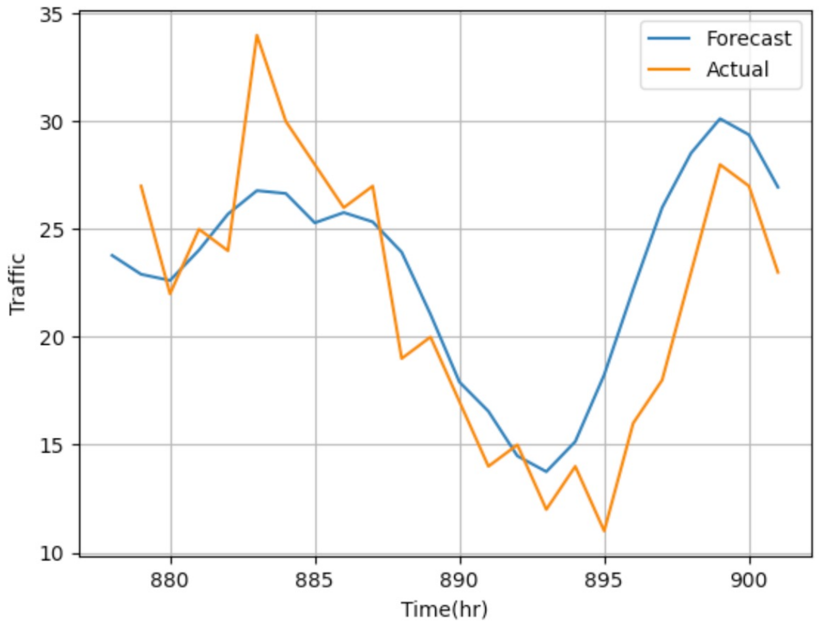
LSTMs expects an input which has sequence of features, hence is a dependable algorithm in time series prediction

Implementation: Dataset 1

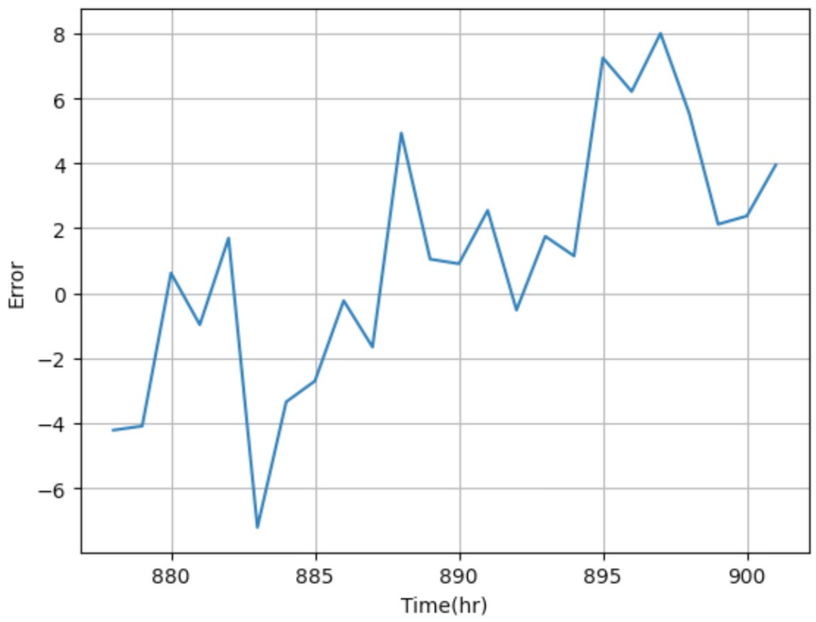
Python Modules used to train the model	Library for the Model: statsmodels.tsa.arima.model Model imported : ARIMA Method to train the Model : model.fit() {wrt the training dataset}
Python Modules used to test the model	Predicted values obtained from : model.forcast (steps) To calculate error: np.sqrt((error**2).mean())
Hyperparameter details	Order of the ARIMA : (p=30,d=0,q=1)

Experimental Results: Dataset 1

1) The Model was trained with 878 samples
2) Forecast was made for the next 24 hours
3) Model was tested using MSE loss metric



Comparison of predicted values of the Model and the actual model



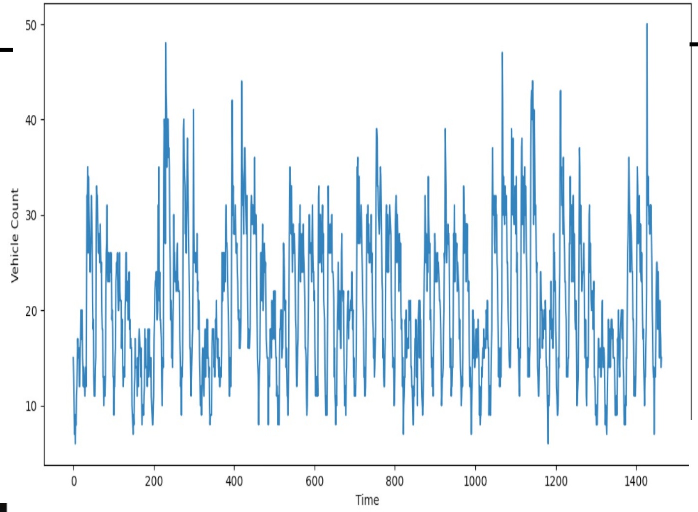
Error at each time instant

Model MSE Score
3.878
Inference: forecast of the density is off by 4 cars

Hyperparameters used to obtain these results are:
(p=30,d=0,q=1)

Inference and Observations: Dataset 1

Dataset



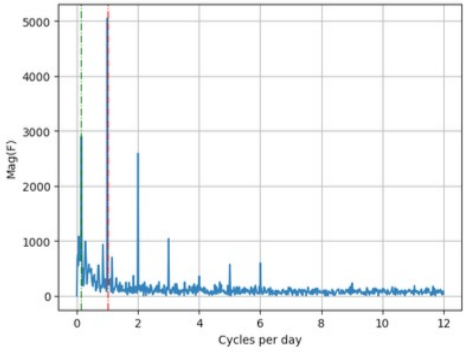
Dataset was predicted as not seasonal with Augmented Dickey–Fuller test (ADF) test

```
ADF Statistic: -3.6707831962784936
p-value: 0.004543396291786597
Critical Values:
   1%: -3.4349024693573584
   5%: -2.8635506057382325
   10%: -2.5678404322793846
Data is stationary
```

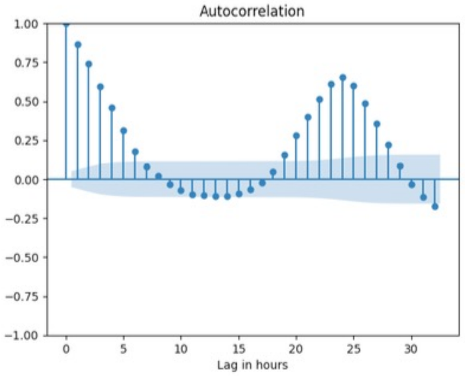
Dataset is Seasonal (verified by FFT and Autocorrelation)

FFT parameters: X: cycles per day, Y: Mag(F)

Autocorrelation parameters: Lag in hours



(a) Fourier Transform

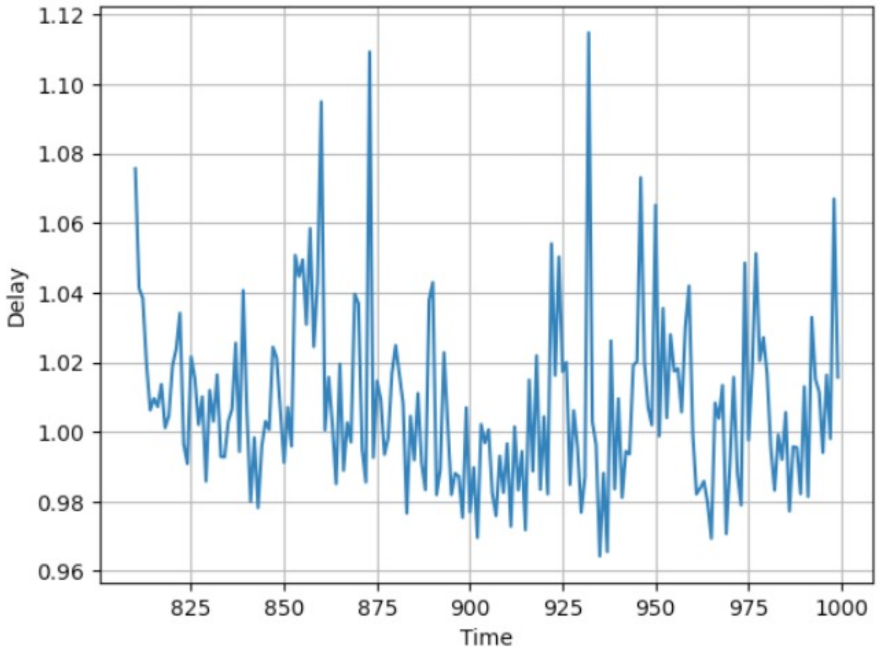


(b) Auto-correlation

Implementation: Dataset 2

Python Modules used to train the model	Library for the Model: sklearn.ensemble Model imported: RandomForestRegressor Method to train the Model : RandomForestRegressor.fit(n_estimators)
Python Modules used to test the model	Predction was done using : RandomForestRegressor.predict() Valuation Metric: {Mean Squared error} Calculated using: MeanSquaredError(X,X_pred)
Hyperparameter details	Number of regressors= 100

Experimental Results: Dataset 2



Delay Estimates for 200 samples

1) The model was trained for 800 samples and tested for 200 samples
2) MSE Loss was considered as the scoring metric.

Model MSE score

0.00485

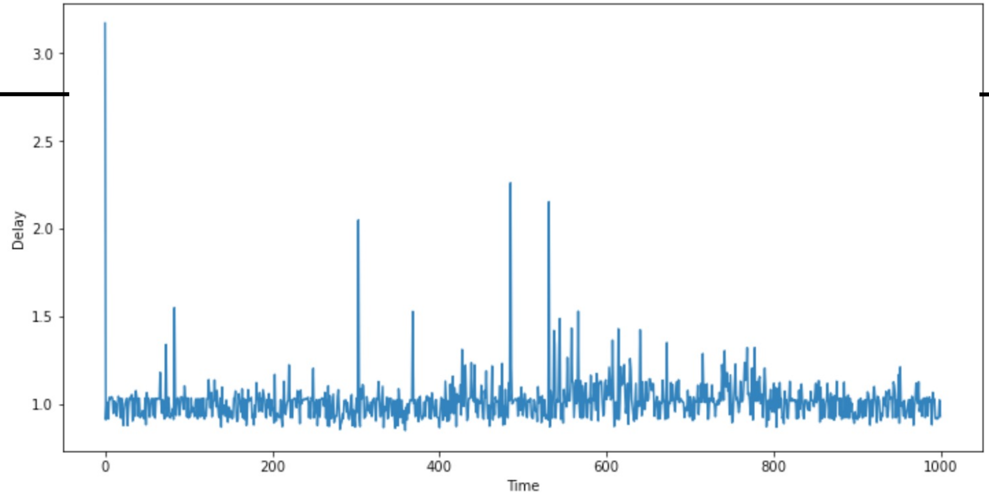
Inference: Very close to the optimal model

Hyperparameters for this model

Number of regressors=100

Inference and Observations: Dataset 2

Dataset



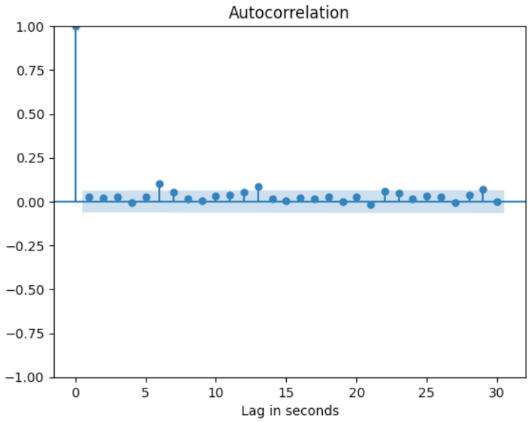
Dataset was predicted as stationary (not seasonal) with Augmented Dickey–Fuller test (ADF) test

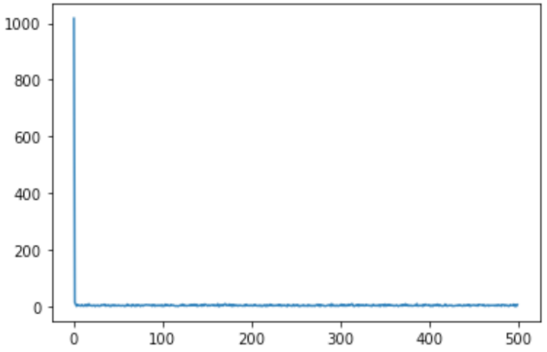
Predictions of the ADF test were validated by FFT and autocorrelation of the data

```
ADF Statistic: -6.196518378910545
p-value: 5.947121735927869 6e-08
Critical Values:
   1%: -3.4369927443074353
   5%: -2.864472756705845
   10%: -2.5683315460972 38
Data is stationary
```

FFT Parameters: **Frequency of delay samples**

Autocorrelation Parameters: **Lag in seconds**



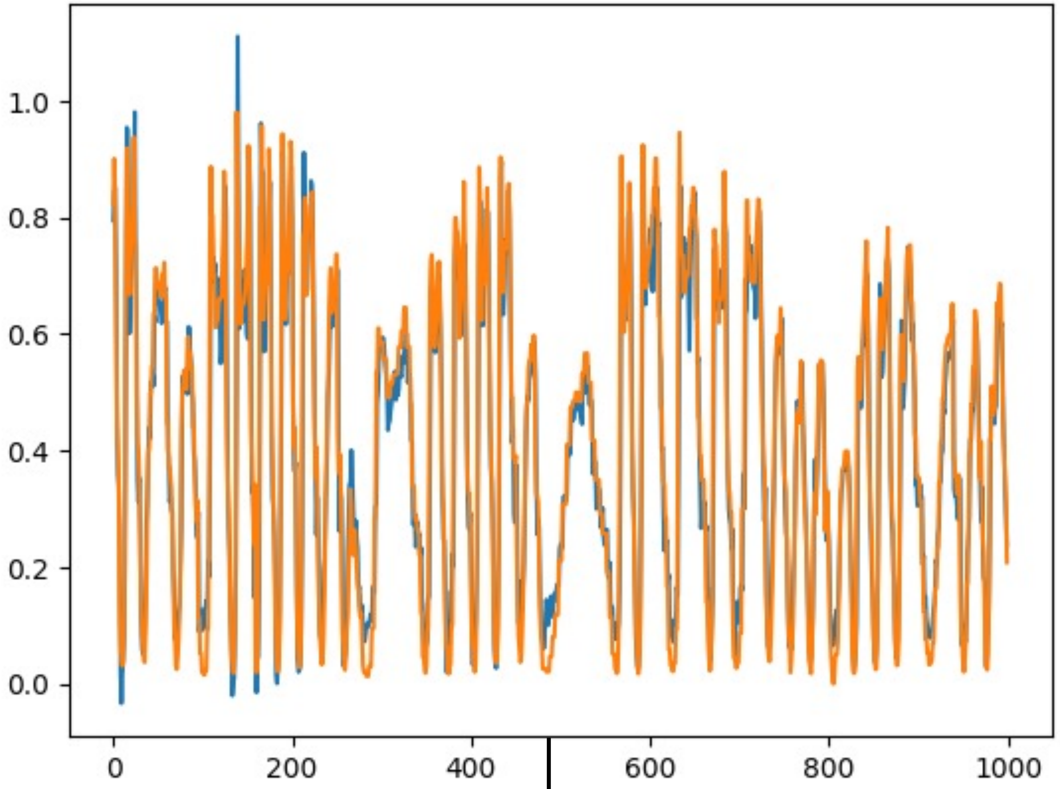


(a) Fourier Transform

Implementation: Dataset 3

Python Modules used to train the model	Library for the Model: from tensorflow.keras.layers Model imported: LSTM Method to train the Model : model.fit(X_train,Y_train) Optimizer Used: ADAM Optimizer
Python Modules used to test the model	Prediction was done using : model.predict(X_test) Valuation Metric: Mean Squared Error Calculated using : model.evaluate(X_test,Y_test)
Hyperparameter details	Number of dense layers: 1 Number of LSTM units : 64

Experimental Results: Dataset 3



Predicted v/s actual traffic densities
Note: all the parameters in this dataset have been normalized

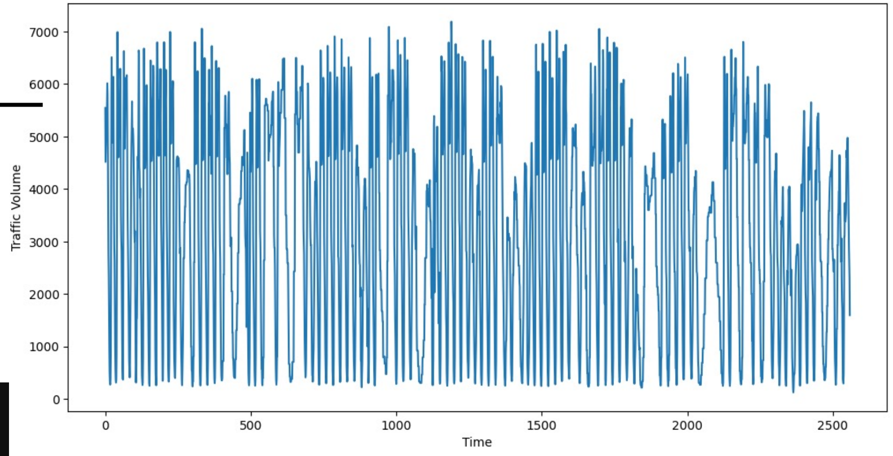
1) The Model was trained with 1535 samples
2) Model was tested using MSE loss metric

Model MSE score
0.005932
Inference: LSTM is learning the curve accurately.

Hyperparameters used to obtain these results are:
(num_layers=1)
(num_LSTMs=64)

Inference and Observation: Dataset 3

Dataset



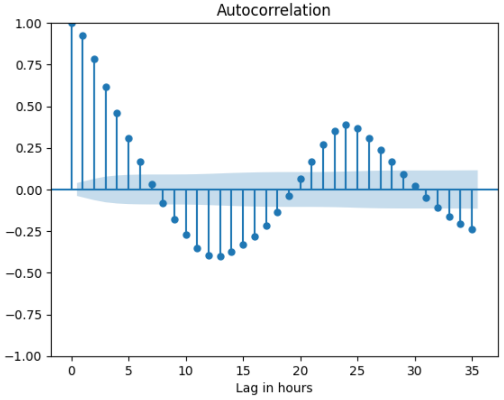
Dataset was predicted as stationary by ADF test

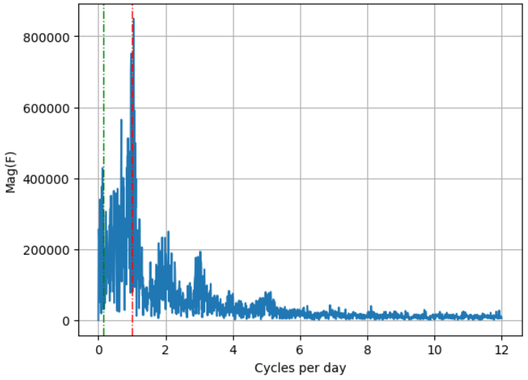
```
ADF Statistic: -10.215246602059066
p-value: 5.50341246368357e-18
Critical Values:
   1%: -3.43293324257297
   5%: -2.8626812695784225
   10%: -2.5673775408540145
Data is stationary
```

The Predictions made by the ADF test was verified by FFT and autocorrelation

FFT parameters: X: cycles per day, Y: Mag(F)

Autocorrelation parameters: Lag in hours





Conclusion

- Time series datasets were considered to learn the delay estimates in a mobile edge computing environment.
- ARIMA, RFR, LSTM algorithms were implemented on Datasets 1,2,3 respectively.
- Various features and performance of the input and response of the model were analyzed using graphs, transforms and metrics.
- To best of our Knowledge, LSTM performed the best in learning the trends of the time series dataset provided to it.


DEMO

QNA
Thanks